

Technická univerzita v Liberci

Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: B 2612 – Elektronika a informatika

Studijní obor: 2612R011 – Elektronické informační řídicí systémy

1802R022 – Informatika a logistika

Ovládání digitálních fotoaparátů pomocí Delphi

Remote control cameras in Delphi

Bakalářská práce

Autor: **Václav Stodůlka**

Vedoucí práce: Ing. Tomáš Pluhař

Konzultant: Ing. Přemysl Svoboda

Prohlášení

Byl (a) jsem seznámen (a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé BP a prohlašuji, že s o u h l a s í m s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom(a) toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem

Datum

Abstrakt:

Cílem této bakalářské práce je seznámit čtenáře se vzdáleným ovládáním fotoaparátu pomocí programovacího jazyku Delphi. Výsledkem práce je ucelený návod jak použít nástroje od výrobců fotoaparátů. Práce se snaží nebýt pouhým přepisem volně dostupných manuálů, ale spíše postřehů ke vzdálenému ovládání fotoaparátů.

Abstract:

The aim of this bachelor assignment is to introduce to readers a remote control for a camera with the ability of programming language Delphi. The result of this assignment is a guideline for using software development kit (SDK) from manufacturers of digital cameras. The assignment is not only attempting to transcript freely reachable manuals, but rather to provide tips for remote control of digital cameras.

Klíčová slova:

Vzdálené řízení, digitální fotoaparát, Delphi, SDK vývojové nástroje

Keywords:

Remote control, digital camera, Delphi, software development kit (SDK)

Obsah

PROHLÁŠENÍ.....	3
ABSTRAKT:.....	4
1 ÚVOD.....	6
2 DIGITÁLNÍ FOTOAPARÁTY	7
2.1 OBECNĚ O DIGITÁLNÍCH FOTOAPARÁTECH.....	7
2.2 TŘÍDY DIGITÁLNÍCH FOTOAPARÁTŮ	7
2.2.1 Fotomobily.....	7
2.2.2 Stylové kompakty.....	7
2.2.3 Vyspělé kompakty.....	8
2.2.4 Elektronické zrcadlovky (EVF)	8
2.2.5 Digitální zrcadlovky.....	8
2.3 PRINCIP ČINNOSTI	8
2.4 RAW FORMÁT	9
2.4.1 Standardizování RAW.....	9
2.4.2 Zobrazení RAW.....	10
2.4.3 Postup zpracování RAW.....	11
2.4.4 Bayerova interpolace.....	12
2.5 FIRMWARE	13
3 VZDÁLENÉ ŘÍZENÍ FOTOAPARÁTŮ (REMOTE CONTROL)	14
3.1 PODPORA	14
3.2 SOFTWARE DEVELOPMENT KIT (SDK)	17
3.2.1 Problémy.....	18
4 OLYMPUS SDK	19
4.1 DISTRIBUCE	19
4.2 INSTALACE.....	19
4.3 OLYMPUS SDK V DELPHI 7	19
4.4 NALEZENÍ A ZOBRAZENÍ PŘIPOJENÝCH FOTOAPARÁTŮ.....	20
4.5 POŘÍZENÍ FOTOGRAFIE	20
4.6 OPTICKÝ / DIGITÁLNÍ ZOOM	21
4.7 NÁHLED Z FOTOAPARÁTU	22
4.8 OSTATNÍ FUNKCE	22
4.9 NADSTANDARDNÍ FUNKCE	23
4.10 DETEKCE CHYB	24
4.11 KONSTANTY.....	24
5 CANON SDK.....	25
5.1 DISTRIBUCE	25
5.2 POUŽITÍ.....	25
5.3 STRUKTURA	26
5.4 CANON SDK V DELPHI	28
5.5 ZAČÁTEK KOMUNIKACE	29
5.6 POVOLENÍ/ZAKÁZÁNÍ ZÁSAHU UŽIVATELE.....	30
5.7 POŘÍZENÍ FOTOGRAFIE	31
5.8 OSTATNÍ FUNKCE	32
5.9 PRÁCE S FOTOGRAFIÍ.....	33
5.10 DETEKCE CHYB	35
6 ZÁVĚR.....	36
6.1 SDK OLYMPUS	36
6.2 SDK CANON	36

1 Úvod

Digitální fotoaparáty jsou dnes již natolik rozšířené a cenově dostupné, že si je může dovolit téměř kdokoli. Technologie, která není zatím masově známá a která se velice rychle vyvíjí je vzdálené řízení (tzv. Remote Control). Jedná se o způsob ovládání fotoaparátu, kdy fotograf nemusí být přímo u fotoaparátu. Možností jak vzdáleně ovládat fotoaparát je více, tato práce se zabývá tou nejnovější – vzdálené řízení pomocí počítače.

Práce je rozdělena do několika částí. První část popisuje obecně digitální fotoaparáty. Druhá část se věnuje vzdálenému ovládání fotoaparátů. Další části jsou věnovány vždy jednomu výrobcí fotoaparátů, který podporuje vzdálené ovládání. Poslední část zhodnocuje celou práci a vysvětluje vhodnost a nevhodnost použití programovacího jazyka Delphi.

2 Digitální fotoaparáty

2.1 Obecně o digitálních fotoaparátech

Digitální fotoaparát je fotoaparát, zaznamenávající obraz v digitální formě, takže může být okamžitě zobrazen na zabudovaném displeji, nebo nahrán do počítače. V roce 2006 digitální fotoaparáty na trhu dominují.

Základní funkcí digitálního fotoaparátu je snímání statických obrazů do podoby tzv. digitální fotografie a umožnit tak jejich další zpracování, např. pomocí běžného počítače, jejich tisk či vyvolání speciální osvitovou jednotkou do výsledné podoby jako u klasické fotografie.

Dnešní digitální fotoaparáty nabízí kromě své základní funkce také řadu dalších doplňujících a rozšiřujících funkcí, které souvisejí ať už přímo či nepřímo se zpracováváním obrazovanými daty. Některé fotoaparáty tak dokážou kromě obrazu zaznamenat i pohyblivé scény ve formě videa nebo zvukový záznam ve formě ozvučeného videa, nebo přidávat poznámky k pořízeným snímkům.

2.2 Třídy digitálních fotoaparátů

2.2.1 Fotomobily

V pravém slova smyslu se nejedná o fotoaparáty. Schopnost fotografovat není prioritou těchto zařízení a vestavěný fotoaparát je jen funkcí navíc. Jednoduchá optika, jednoduché ovládání, ultraminiaturní senzor – to vše zatím nedává předpoklady pro kvalitní fotografii schopnou tisku na papír velikosti alespoň 13x18cm. Proto nejsou používány k pořizování památních fotografií, ale pouze k pohotovému fotodokumentaci např. otevíracích hodin obchodů a jiných.

2.2.2 Stylové kompakty

I když účelem těchto zařízení je již fotografie, tak je u nich kladen spíše důraz na image než na praktické stránky focení. Jedná se většinou o plně automaty s jednoduchou optikou a důmyslným designem. Designu je podřízeno vše a tak je skoro pravidlem velmi malý senzor a miniaturní objektiv s extrémně krátkou ohniskovou vzdáleností. Důsledkem krátkých ohniskových vzdáleností je extrémní hloubka ostroty (vše je stále ostré) a tak často zcela chybí ostřicí systém. To vyhovuje jak konstruktérům, tak majitelům – téměř totiž nelze pořídit nezaostřenou fotografii. Za dobrých světelných

podmínek jsou tyto kompakty schopné pořídit i kvalitní fotografie. Řízení expozice, hloubky ostrosti, použití filtrů či rychlá reakce na spoušť však u těchto fotoaparátů nejsou dostupné.

2.2.3 Vyspělé kompakty

Tato třída digitálních fotoaparátů patří stále do kategorie kompaktních, ale žebříček priorit je nastaven jinak. V první řadě je to již možnost kvalitní kreativní či umělecké fotografie, teprve v druhé řadě jsou rozměry, váha, cena a jednoduchost. Senzor bývá větší a je nastaven na kompromis kvalita obrazu / přijatelný rozměr fotoaparátu. Větší senzor vyvolá současně potřebu objektivu s delším ohniskem, kde jednak stoupá kvalita a současně je již možné pracovat s hloubkou ostrosti. Také bývá větší zoom rozsah (typicky 8 – 12x). Fotoaparáty již nabízejí celou řadu příslušenství – filtry na objektiv, externí blesky, drátové spouště ale také možnost rozsáhlého nastavení parametrů a veličin. Stoupá i rychlost reakce na spoušť umožňující reportážní snímání. Bohužel tím vším stoupá i cena, rozměr a váha fotoaparátu.

2.2.4 Elektronické zrcadlovky (EVF)

Elektronická zrcadlovka představuje jakýsi digitální mezikrok mezi kompaktem a pravou zrcadlovkou. Fotoaparáty mají v hledáčku displej, ukazující obraz ze senzoru. Moderní EVF přístroje mají rozlišení okolo devíti megapixelů a zoom s objektivem s velkým rozsahem ohniskových vzdáleností. Většina těchto fotoaparátů má také optický stabilizátor obrazu.

2.2.5 Digitální zrcadlovky

Digitální zrcadlovka je v podstatě běžná zrcadlovka, která místo filmu obsahuje CCD nebo CMOS senzor. Z principu konstrukce vyplývá maximální věrnost zobrazení v hledáčku, možnost použití výměnných objektivů a nemožnost natáčet videosekvence nebo používat displej ke kompozici záběru. Druhé omezení se ale podařilo prorazit Olympusu, jehož zrcadlovka E-330 má v hranolu další senzor speciálně pro ten účel.

2.3 Princip činnosti

Princip digitálního fotoaparátu vychází z konstrukce klasického fotoaparátu. Jádrem je světlocitlivá plocha snímače na bázi technologie CCD nebo CMOS. Na plochu senzoru je promítán obraz přes systém optických čoček v objektivu. Světelná

energie, která přichází ze snímaného prostoru (scény), je v jednotlivých pixelech (obrazových bodech) převáděna na elektrický signál a uložena v podobě vázaného náboje (u technologie CCP). Náboj vzniká postupně během expozice čipu, kdy je otevřena uzávěrka fotoaparátu a světlo může dopadat na čip. Princip vzniku elektrického náboje je založen na fotoelektrickém jevu s tím rozdílem, že náboje neodtékají okamžitě do vnějšího obvodu, ale jsou izolovány v nábojových zásobnících v elektricky izolované struktuře čipu.

Po uzavření uzávěrky jsou vygenerované náboje z čipu postupně odváděny a měřeny speciálním zesilovačem pro každý jednotlivý pixel. Takto získaný signál je dále převeden AD převodníkem na signál v binárním kódu. Vzniklý datový proud je pak pomocí mikroprocesoru různě upravován a převeden do některého grafického formátu používaného pro záznam obrazových dat (RAW, JPEG nebo TIFF). Výsledný datový soubor je uložen zpravidla na paměťové medium.

2.4 RAW formát

Obecně řečeno obrazový RAW formát obsahuje jen minimálně zpracovaná data ze senzoru fotoaparátu. Název byl vytvořen z anglického slova raw (vyslovováno *ró*) znamenající surový, neupravený, hrubý. RAW je často nazýván digitálním negativem, i když s negativním typem obrazu nemá nic společného. Název vznikl jen z podobnosti, že RAW formát je třeba ze surových dat v počítači také „vyvolat“, podobně jako se musí chemicky vyvolat klasický filmový negativ. Proto se také často hovoří o vyvolání RAWu, což ale znamená výpočet obrazu v PC. RAW formát není jen doménou fotoaparátů, ale nabízí ho i některé scannery, které jsou schopné také poskytnout surová data a nechat jejich zpracování až na uživateli u PC.

2.4.1 Standardizování RAW

RAW formát není bohužel nijak standardizován a výrobci si každý podle svých potřeb, preferencí a použité technologie (senzoru) definovali vlastní RAW formát. Rozdíly jsou značné, dokumentace k jednotlivým RAW formátům také není většinou dostupná. Je tak třeba používat programy určené pro konkrétní značku a často i pro konkrétní model. Výrobci navíc částečně pod vlivem technologického vývoje, částečně

jako obchodní politiku, RAW formát model od modelu mění. Nikon své RAWy dokonce šifruje, aby k nim neautorizované programy neměly přístup.

Bylo již několik pokusů o standardizaci RAW formátu. Asi nejvýznamnější pokus učinila firma Adobe, která definovala svůj otevřený a standardní formát DNG (Digital Negativ) a nabídla ho výrobcům digitálních fotoaparátů a jiných zařízení. Zatím ale výrobci tento formát nepřijali. Nestandardnost RAW formátu dokládá tabulka Tab. 1

Tab. 1 - Typ RAW podle výrobce

Přípona	Výrobce
.raf	Fuji
.crw .cr2	Canon
.tif .kdc .dcr	Kodak
.mrw	Minolta
.nef	Nikon
.orf	Olympus
.dng	Adobe – otevřený standard
.ptx .pef	Pentax
.arw .srf .sr2	Sony
.x3f	Sigma
.erf	Epson
.mef .mos	Mamiya
.raw	Panasonic

2.4.2 Zobrazení RAW

RAW soubor primárně obsahuje hrubé digitální data toho, co uviděl senzor, a současně data sdělující okolnosti pořízení snímku (metadata). RAW data tedy nejsou ukotvena k žádnému standardnímu barevnému prostoru, v datech není vyvážena bílá, není definován bílý bod ani gamma křivka atd.

Na rozdíl od JPEGu si RAW fotografii na PC nemůžete ani zběžně prohlédnout. Pokud totiž neproběhne Bayerova interpolace, snímek de facto neexistuje. Na trhu však existuje řada pokročilejších prohlížečů, které jsou RAW obrázky schopné alespoň zběžně zobrazit. K zobrazení RAW snímku používají dvě strategie.

Jednodušší strategie je použít a zobrazit malý náhledový JPEG soubor, který ve většině RAW souborů bývá zabalen. Je to malý náhled o rozlišení cca 300x200 pixelů (tzv. thumbnail), který mj. slouží k tomu, aby bylo možné si prohlížet snímky na displeji fotoaparátu a fotoaparát je nemusel pokaždé vypočítávat z RAW.

Složitější strategie je obrázek z RAW vypočítat a zobrazit tedy úplnou fotografii. Někteří výrobci fotoaparátů proto uvolňují dekodovací programy, které umožní RAW snímek zobrazit. Nejedná se tedy o plnohodnotné vyvolání RAWu se všemi možnými parametry, ale o kvalitní zobrazení většinou jen se standardním nastavením. I to je však pro zobrazení a prohlížení RAW snímků na PC (např. při jejich třídění) velká pomoc.

2.4.3 Postup zpracování RAW

Fotoaparát po stisknutí spouště získá ze senzoru de facto RAW. Obrazový procesor jej však bleskurychle „vyvolá“ (vypočítá) a uloží na kartu již jako běžný JPEG. Z toho vyplývá, že všechny parametry, které je možné při vyvolání RAWu nastavit v PC (u specializovaných programů), musí fotoaparát nějak nastavit také. Jinak by totiž nemohl vzniknout finální JPEG. Zpracování RAW zobrazuje obrázek Obr. 1



Obr. 1 - Zpracování RAW

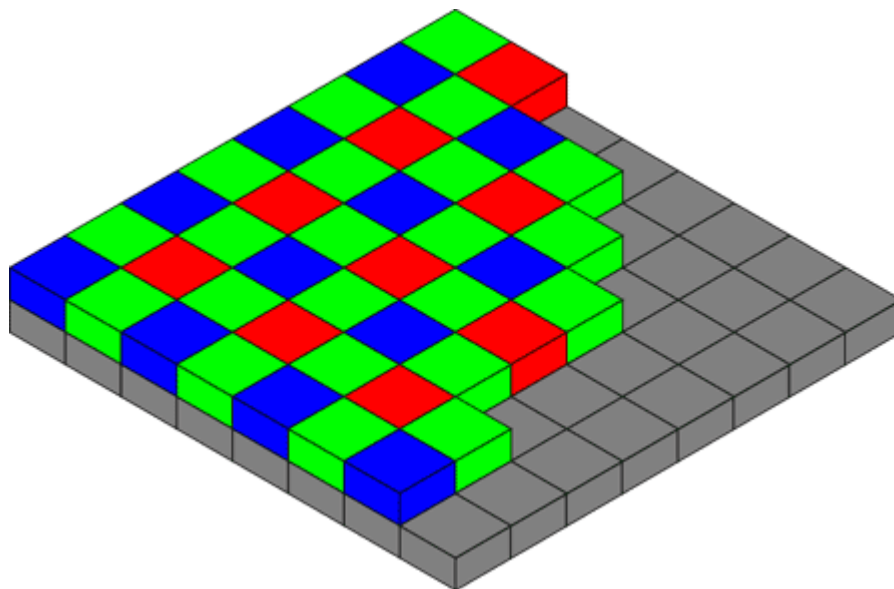
Červené boxy znázorňují procesy elektroniky senzoru a na jejich konci je právě RAW. K získání reálné fotografie je nutné, aby proběhla jednak Bayerova interpolace, jednak byla vyvážena bílá a také aby byly nastaveny všechny další parametry obrazu (modré boxy). Pokud se fotografie ukládá ve formátu JPEG, tak je nutné ještě nastavit a provést JPEG kompresi (zeleně). Řada parametrů je tak nutně ve fotoaparátu nastavena pevně. Některé sice lze ovlivňovat v menu (například vyvážení bílé, kontrast, sytost

barev, míru doostření atp.), ale vždy před snímkem, což je de facto naslepo. Opravit ve fotoaparátu již vyvolaný snímek by sice teoreticky bylo možné, ale fotoaparáty tuto funkci nenabízejí, protože měnit parametry vyvolávacího RAW procesu podle náhledu na malém zadním displeji by bylo stejně nepoužitelné.

2.4.4 Bayerova interpolace

Z technologického hlediska je obtížné (ba nemožné) vytvořit buňku senzoru, která by viděla rovnou barvu v lidském slova smyslu. Buňky senzoru jsou tedy barvoslepy a zaznamenávají jen množství světla, tedy jas. Aby senzor vytvořil dojem, že vidí barevně, má nad svými buňkami různě barevná skla (filtry) uspořádané v pravidelné struktuře. Různí výrobci používají různé barvy, nicméně masově používané jsou odstíny červené (Red), zelené (Green) a modré (Blue).

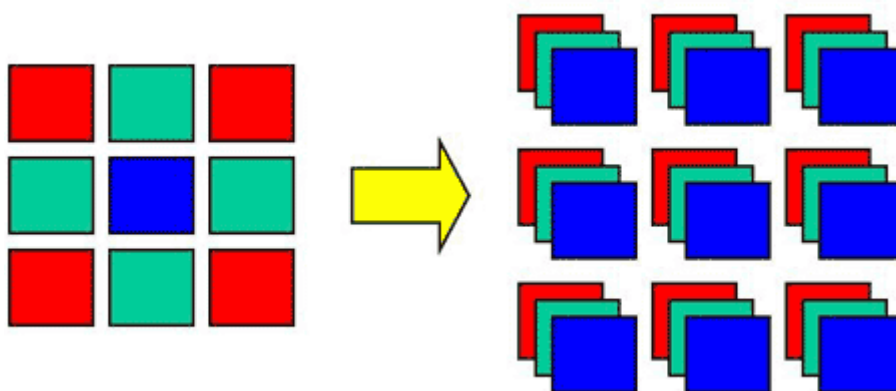
12 MPix fotoaparát má tedy skutečně senzor mající 12 milionů buněk, ale jen barvoslepých. Nad 3 miliony z nich je červený filtr, nad 3 miliony z nich modrý a nad 6 miliony zelený. Dvojnásobné množství „zelených“ buněk vyplývá z lidského oka, které je také na zelenou barvu nejcitlivější. Barevná skla nad buňkami senzoru jsou uspořádána do pravidelné struktury, které se říká Bayerova maska Obr. 2.



Obr. 2 - Bayerova maska

Bayerova maska nad senzorem je systém pravidelně uspořádaných barevných filtrů. Díky tomu senzor měří v jedné čtvrtině buněk červenou barvu, v další čtvrtině buněk modrou a v polovině zelenou. Výsledná fotografie je až produktem následného výpočtu. Z výše uvedeného vyplývá, že žádná buňka senzoru nevidí úplnou a skutečnou barvu.

Vidí vždy buď červenou část, zelenou nebo modrou. Barvu jednoho pixelu fotografie je tak potřeba vypočítat s tím, že se použijí okolní pixely pro získání ostatních barev. Tomuto procesu se říká Bayerova interpolace (Obr. 3) a její princip je de facto totožný ve všech digitálních fotoaparátech a scannerech.



Obr. 3 - Bayerova interpolace

Bayerova interpolace je způsob, jak spočítat plnohodnotnou informaci pro každý pixel z okolních a různě barevně citlivých buněk senzoru. Je to matematický výpočet.

2.5 Firmware

Firmware je programové vybavení, které je součástí elektronického zařízení. Tyto programy jsou většinou naprogramovány (vypáleny) do nevolatilních pamětí (ROM, EPROM, ...) firmou vyrábějící dané zařízení. Do firmware bychom měli počítat také obsah programovatelných hradlových polí, pokud jsou v daném zařízení použita. V závislosti na způsobu uložení jej lze v některých případech modifikovat buď výměnou, nebo přepsáním obsahu. Firmware bývá často úzce vázaný na hardware, pro který je určen.

Typickým příkladem zařízení obsahujících firmware jsou mobilní telefony a kapesní počítače. Avšak v dnešní době obsahuje drtivá většina elektronických zařízení nějaký programový kód - firmware.

3 Vzdálené řízení fotoaparátů (Remote control)

Od nepaměti pociťoval každý pokročilejší fotograf potřebu ovládat svůj přístroj na dálku. Důvody mohou být různé. U rodinné fotografie je to například to, že chce být fotograf se svými blízkými také na snímku. U focení ze stativu zase hraje roli, že stisknutí spouště může nepatrně rozechvět fotoaparát což pak má vliv na výslednou kvalitu fotografie.

Typů remote control je hodně. Od dálkových mechanických spouští až po ovládání fotoaparátu přes počítač – tomuto typu remote control se tato práce věnuje.

3.1 Podpora

Protože remote control je technologie relativně nová podpora pro nižší třídy fotoaparátů (kompakty) neexistuje. I u zrcadlovek se nesetkáváme s velkou podporou, nejlépe propracované vzdálené řízení má Olympus, poté Canon a Nikon. Ostatní výrobci vzdálené řízení nepodporují vůbec, nebo jen minimálně.

Podporu pro fotoaparáty Olympus vyjadřuje tabulka Tab. 2.

Tab. 2 - Podpora remote control u fotoaparátů Olympus

Aktuální modely	Starší modely		
	Verze 3.4	Verze 2.0	Verze 1.5
E-3	C-8080 Wide Zoom	D-360L/C-860L	D-220L/C-420L
E-510	C-7070 Wide Zoom	D-460Z/C-960Z	D-320L/C-820L
E-410	C-7000 Zoom / C-70 Zoom	D-490Z/C-990Z	D-340L/C-840L
E-500	C-5060 Wide Zoom*	C-211 Zoom	D-340R/C-830L
E-400	C-5050 Zoom*	C-2100 Ultra Zoom	D-400Z/C-900Z
E-330	C-4040 Zoom	C-3000 Zoom	D-450Z/C-920Z
E-300	C-4000 Zoom / C-4100 Zoom	C-3030 Zoom	C-2000 Zoom
E-1	C-3040 Zoom		C-2020 Zoom
	C-3020 Zoom / C-3100 Zoom		D-500L/C-1000L
	C-2040 Zoom		D-600L/C-1400L
	C-765 Ultra Zoom		D-620L/C-1400XL
	C-760 Ultra Zoom		
	C-750 Ultra Zoom*		
	C-740 Ultra Zoom*		
	C-730 Ultra Zoom		
	C-700 Ultra Zoom		
	D-40 Zoom / C-40 Zoom		
	C-21		
	E-100 RS		
	SP-700		
	SP-500 Ultra Zoom*		
	SP-350 Zoom*		
	FE-130/X-720		
	D-595/C-500 Zoom		

* Tento fotoaparát vyžaduje před používáním remote control aktualizaci firmwaru. Tento software je dostupný ze serveru Olympus (URL: <http://www.olympus.co.jp/en/support/msgs/digicamera/download/software/#firmware>)

Podporu pro fotoaparáty Canon vyjadřuje tabulka Tab. 3.

Tab. 3 - Podpora remote control u fotoaparátů Canon

Pro series	G-series	S-series	A-series	Digital IXUS
Pro90 IS	G1	S30	A10	SD100 (Digital IXUS II)
Pro 1	G2	S40	A20	SD110 (Digital IXUS lis)
	G3	S45	A30	S230, Digital IXUS 330
	G5	S50	A60	S400 (Digital IXUS 400)
	G6	S60	A70	S410 (Digital IXUS 430)
	G7	S70	A75	S500 (Digital IXUS 500)
		S80	A80	
		S1 IS	A85	
		S2 IS	A95	
		S3 IS	A100	
		S5 IS	A200	
			A310	
			A400	
			A510	
			A520	
			A620	
			A640	

Podporu pro fotoaparáty Nikon vyjadřuje tabulka Tab. 4.

Tab. 4 - Podpora remote control u fotoaparátů Nikon

D100	D200
D1	D3
D1X	D300
D1H	D50
D2H	D70/D70s
D2Hs	D80
D2X	D40
D2Xs	D40x

3.2 Software Development Kit (SDK)

SDK je obecně sada vývojových prostředků pro vývojáře, určený k vytváření aplikací pro určitý softwarový balík, hardware, počítačový systém, konzoly pro videohry, operační systém nebo podobné platformy.

Může to být něco jednoduchého, jako aplikační rozhraní ve formě souborů kompatibilních s určitým programovacím jazykem, nebo může zahrnovat hardware pro komunikaci se systémem. Běžné nástroje zahrnují ladicí pomůcky a jiné nástroje často používané v IDE. SDK většinou obsahuje i ukázkový kód s poznámkami, nebo další technickou dokumentaci.

SDK bývá často dostupné přes internet. Mnoho SDK bývá zdarma pro vyzkoušení daných nástrojů a programovacího jazyka. Většinou bývá ale vázáno licenci.

Dodavatelé SDK pro určité systémy se mohou pomocí těchto kitů vyhnout aktualizaci rozsáhlého softwaru. Například Microsoft i Apple dodávají Driver Development Kits (DDK) pro aktualizaci ovladačů zařízení.

SDK pro fotoaparáty obsahuje knihovny potřebné pro komunikaci počítače s fotoaparátem, dokumentaci a přepsané knihovny do tříd, které stačí přidat do

vytvářeného projektu. Tyto třídy jsou ale jen pro jazyk C++ (pouze Olympus má třídy přepsány do Delphi).

Protože se SDK pro jednotlivé výrobce liší, tato práce se jim věnovat odděleně.

3.2.1 Problémy

Používáním vzdáleného řízení se uživatel může setkat s různými problémy, které vznikají neznalostí, nebo špatným zacházením.

Pokud je fotoaparát přepnutý do remote control módu, není standardním způsobem možné stahovat fotky do počítače, program pro ovládání v sobě musí mít zabudován modul pro stahování fotografií.

Některé fotoaparáty Olympus nemají standardně vzdálené řízení povoleno, proto se musí povolit ručně. Postup se liší v závislosti na modelu fotoaparátu.

U modelů Olympusu C-2100UZ, C-3000Z, C-3030Z, D-360L, D-460Z, D-490Z a C-211Z se musí při zapnutém fotoaparátu odpojeném od počítače přepnout do módu "A/S/M".

U modelů C-2040Z, C-3020Z, C-3040Z, C-40Z/D-40Z, C-4040Z a E-100RS se musí nejprve odpojit fotoaparát od počítače, nastavit mód "A/S/M", zapnout fotoaparát, otevřít dvířka slotu pro Smart Media karty, na 3 sekundy současně stisknout tlačítko aktivující menu ("OK/Menu") a tlačítko pro zapnutí/vypnutí LCD displeje. Na displeji se objeví servisní menu, ve kterém se přepne „Camera Control“ na „ON“. Nyní stačí volbu potvrdit a zavřít dvířka pro Smart Media kartu. Tento postup se dá použít i pro starší fotoaparáty.

U modelu C-700UZ je postup trochu odlišný. Fotoaparát odpojíme od počítače, nastavíme mód "A/S/M", zapneme fotoaparát, vyndáme Smart Media kartu (stále při zapnutém fotoaparátu!), zavřeme dvířka od Smart Media slotu, na 3 sekundy současně stiskneme tlačítko "OK/Menu" a tlačítko aktivující LCD displej. Opět se objeví servisní menu, ve kterém nastavíme „Camera Control“ na „ON“ a můžeme vrátit zpátky Smart Media kartu a zavřít dvířka.

4 Olympus SDK

Podrobnosti všech instrukcí z SDK Olympus jsou k nalezení v manuálu na přiloženém CD (SDK Olympus\ SDK 3.4 Reference Guide.pdf). Program na přiloženém CD byl vyvíjen a testován na fotoaparátu Olympus C-765 Ultra Zoom.

4.1 Distribuce

SDK pro fotoaparáty Olympus je distribuováno pomocí webové stránky (URL: <http://developer.olympus.com/cameras.html>), kde jsou veškeré informace o SDK. Po registraci a přihlášení se můžou stáhnout veškeré potřebné soubory k ovládání fotoaparátu a je dostupné také oficiální fórum. Registrace je zpoplatněna.

4.2 Instalace

Instalace je velice jednoduchá. Provádí se spuštěním souboru setup.exe (na přiloženém CD ve složce „SDK Olympus“ ve stejné složce je i registrační klíč potřebný k dokončení instalace. Instalační program nakopíruje do systému veškeré potřebné knihovny. Posledním krokem před psaním programu je přidání záznamu do registru Windows, odtud potom SDK vyhledává dostupné fotoaparáty. Záznam se přidá pomocí programu, který je součástí instalace, při výchozím nastavení ve složce „Program Files/Olympus/Olympus Developer Program/Digital Camera SDK“. Program se jmenuje „SDKReg.exe“ po jeho spuštění se vybere typ fotoaparátu a uloží se do registrů. Pokud v seznamu není daný typ fotoaparátu uveden a podporuje vzdálené řízení, nemusí se do registrů přidávat. Ve stejné složce se dále nachází soubor „ryecommon.h“, ten obsahuje licenci k používání SDK, dále adresáře, které obsahují potřebné třídy k programování a dokumentaci. Všechny tyto třídy jsou ale pro jazyk C++. Tyto třídy jsou ale přepsány i do Delphi a jsou ke stažení z vývojového centra Olympus (URL: <http://developer.olympus.com/cameras.html>).

4.3 Olympus SDK v Delphi 7

Nejjednodušším způsobem jak komunikovat s fotoaparátem je přes ActiveX komponentu, kterou musíme nejprve přidat do palety komponent Delphi. Uděláme to přes „Component -> Import ActiveX Control...“. Ze seznamu vybereme „RyeNV 1.0 ATL Type Library (Version 1.0)“. Tlačítkem „Install...“ danou komponentu nainstalujeme.

Nová komponenta je nyní přístupná přes paletu komponent, v záložce ActiveX s názvem „RyeNVControl“. Přes tuto komponentu jsou dostupné všechny funkce uvedené v dokumentaci a níže popisované.

4.4 Nalezení a zobrazení připojených fotoaparátů

Základním předpokladem je připojený a zapnutý fotoaparát. Před započítím práce s fotoaparátem se musí inicializovat spojení, to se provádí příkazem „*InitControl*“, který vrací kód chyby. Pokud žádná chyba nenastane, vrací 0.

Počet připojených fotoaparátů vrací funkce „*PropCameraCount*“. Každý fotoaparát má svoje identifikační číslo (dále označované jako CamNum), pomocí kterého se k němu přistupuje. Pokud je připojen jen jeden fotoaparát nevznikají žádné problémy. Při připojení více fotoaparátů se musí rozlišit, které CamNum patří kterému fotoaparátu. Proto existuje funkce „*PropConnectCamera[CamNum]*“, která vrací název modelu fotoaparátu podle CamNum. Tyto operace popisuje ukázkový program na přiloženém CD ve složce „přklady/Zjisteni + zobrazeni“.

Příklad má událost „*onClick*“, která uloží do globální proměnné CamCon vybraného fotoaparátu.

4.5 Pořízení fotografie

Na vyfocení fotografie obsahuje SDK přímo funkce „*Capture(CamNum,Immediate)*“. Tato funkce připraví fotoaparát na vyfocení, zaostří, vyfotí fotografii a uloží fotografii do fotoaparátu. Provedení tohoto příkazu zabere víc času a činnost programu je na tuto dobu pozastavena. Takže fotoaparát nemůže přijmout další příkaz.

Operátorem „*Immediate*“ můžeme nastavit, jestli má program čekat na dokončení focení (false), nebo pokračovat normálně v programu (true). Pokud nastavíme true, musíme hlídat, abychom fotoaparátu neodeslali další instrukci.

Abychom s fotoaparátem mohli provádět jakoukoli operaci, musíme se k němu připojit pomocí funkce „*Connect(CamNum)*“, tato funkce vrací, jestli se připojení podařilo nebo nikoliv. Po dokončení operací se musíme od fotoaparátu odpojit a to pomocí funkce „*Disconnect(CamNum)*“. Pokud bychom se neodpojili, nepůjde

fotoaparát vypnout, dokud nebude program ukončen, nebo dokud nebude odpojen propojovací kabel. Tím se komunikace přeruší.

Protože metoda „*Capture*“ uloží fotografii pouze do fotoaparátu, musíme ji stáhnout na disk. Nejprve musíme nastavit ukazatel na poslední fotografii. Tento ukazatel nastavuje funkce „*propCurrentPicture[CamNum]*“ a protože pomocí funkce „*propPicCount[CamNum]*“ zjistíme počet fotografií, stačí ukazatel nastavit na počet fotografií. Dalším krokem je nastavení bufferu na velikost fotografie, kterou zjistíme funkcí „*propPicSize[CamNum]*“ (vrátí velikost fotografie, na kterou ukazuje ukazatel nastavený v předchozím kroku). Pomocí funkce „*GetPicture(CamNum,BufferSize,Buffer)*“ stáhneme fotografii do bufferu. Nyní můžeme fotografii uložit na disk a následně zobrazit v našem programu.

Protože fotografie jsou na disku uloženy s příponou .jpg, pro zobrazení v programu je musíme přetypovat do .bmp.

Uvedený postup ilustruje přiložený příklad v události „*onClick*“ tlačítka „*btnVyfot*“ s bohatým komentářem.

4.6 Optický / digitální zoom

Optický zoom můžeme ovládat pomocí funkce „*PropOpticalZoom[CamNum]*“. Přechtením téhle funkce zjistíme aktuálně nastavený optický zoom. Přiřazením hodnoty nastavíme optický zoom. SDK neobsahuje žádnou funkci na zjištění maximální a minimální hodnoty zoomu. Ukázkový program je laděn pro fotoaparát Olympus C765 – Ultra zoom. Hodnoty zoomu jsem zjistil nastavením vysoké hodnoty a následným čtením aktuální hodnoty (pro horní hranici) a podobně nastavením nízké hodnoty pro dolní hranici. Firmware fotoaparátu obsahuje kontrolu, která nedovolí vysunutí objektivu nad/pod konstrukční mez. Při nastavení hodnoty mimo rozsah funkce nevrací žádnou chybu.

U programu, který by pracoval s více fotoaparáty, by se musel v cyklu zvyšovat/snižovat zoom a jakmile se přestane měnit aktuální, hodnota máme maximální/minimální hodnotu.

4.7 Náhled z fotoaparátu

Náhled (tj. zobrazení toho co je na displeji) se dělají pomocí funkce „*GetPreview(CamNum, BufferSize, Buffer)*“. Tato funkce funguje podobně jako funkce pro stahování fotografií. Je uvedena v kapitole 4.5. Funkce „*GetPreview*“ převezme aktuální náhled z displeje a uloží do fotoaparátu. S touto „fotografií“ můžeme tedy libovolně nakládat. Pokud ji chceme stáhnout do počítače, provedeme to stejně jako v kapitole 4.5. Náhled má velmi malé rozlišení – rozlišení displeje.

Abychom vytvořili dojem náhledu z fotoaparátu, musíme tuto funkci vykonávat v cyklu. Bohužel funkce je vcelku pomalá, takže nejde zajistit zobrazení plynulého obrazu. Pro tyto účely je lepší zapnout displej (popsáno v kapitole 4.8).

4.8 Ostatní funkce

Protože SDK obsahuje mnoho funkcí, které se dle mého názoru ani nepoužijí, proto jsou v tabulce Tab. 5 uvedeny jen ty, co se používají často. V kapitole 4.9 jsou popsány nadstandardní funkce.

Tab. 5 - Ostatní funkce SDK Olympus

Název funkce	Popis funkce
CameraReady(CamNum)	Vrací, jestli je fotoaparát připraven pro přijetí další instrukce. Použitelné u vícevláknových aplikací
PropPicCount (CamNum)	Počet fotografií uložených ve fotoaparátu
PropRemainCount (CamNum)	Počet zbývajících fotografií co jdou uložit do fotoaparátu. Je dobré kontrolovat před vyfocením, jestli zbývá alespoň jedna fotografie.
PropPowerSaveCamera (CamNum); PropPowerSaveCamera (CamNum)= nValue	Určuje, za jak dlouho přejde fotoaparát do úsporného režimu. Hodnotu můžeme číst nebo přiřazovat. Jdou nastavovat hodnoty 30-600s, v 30 vteřinových skocích.

PropLCDBrightness (CamNum); PropLCDBrightness (CamNum) = nValue	Jas displeje. Hodnotu můžeme číst nebo přiřazovat. Nastavovat se dají level1-level7 , výchozí je level4.
PropFocusMode (CamNum); PropFocusMode (CamNum) = nValue	Slouží k nastavování (čtení nastavení) makra. Jsou povoleny 3 hodnoty. Makro zapnuto, vypnuto a automatický režim
PropFlash (CamNum); PropFlash (CamNum) = nValue	Nastavení blesku. Povolené hodnoty: automatický, vynucený, zakázaný a redukce červených očí
PropResolution (CamNum); PropResolution (CamNum) = nValue	Rozlišení výsledné fotografie. Povolených hodnot je mnoho viz [manual]
propIsoSpeed (CamNum); propIsoSpeed (CamNum) = nValue	Rychlost uzávěrky. Povolené hodnoty: automatická, ISO 100, ISO 200, ISO 400
SetLcdMode(CamNum, nLcdMode)	Nastavení displeje. Povolené hodnoty: vypnuto, zapnuto, prohlížení fotografií

4.9 Nadstandardní funkce

Jedná se o funkce, které nejsou dostupné z menu fotoaparátu. Některé uvádím jen ze zajímavosti, některé jsou použitelné. Nadstandardní funkce zobrazuje tabulka Tab. 6.

Tab. 6 - Nadstandardní funkce SDK Olympus

Název funkce	Popis funkce
EraseLast(CamNum)	Smazání poslední fotografie
EraseAll(CamNum)	Smazání všech fotografií

ErasePicture(CamNum)	Smazání fotografie, na kterou je nastaven ukazatel
PropFolderName (CamNum); PropFolderName (CamNum) = FolderName	Název složky, do které se ukládají fotografie ve fotoaparátu. Název téhle složky jde změnit.
GetPictureProp(nCamNum, pPictureProp)	Vrací ucelené informace (velikost, mód, ve kterém byla vyfocena, čas vyfocení...) o fotografii, určené druhým parametrem funkce.
GetPicTagProp(nCamNum, pPicTagProp)	Podobně jako předchozí funkce vrací informace o fotografii, ale rozšířené (jakým bleskem byla vyfocena, jaké bylo nastavení bílé, rychlost uzávěrky ...).

SDK podporuje mnoho dalších funkcí, které jsou popsány v manuálu.

4.10 Detekce chyb

SDK Olympus vrací chyby pomocí výjimek. Pokud tedy chceme chyby ošetřovat (u profesionálních aplikací je to nezbytné) musíme tyto výjimky odchytávat. V Delphi můžeme použít strukturu Try ... Except . Výjimka vrací konstantu (popsány v kapitole 4.11), která určuje text chyby. Výjimku nevyvolá funkce, která vrací chybu standardně např. „InitControl()“ nebo „Connect (CamNum)“.

4.11 Konstanty

Protože většina funkcí SDK vyžaduje zadávat jen povolené hodnoty, byl zaveden systém konstant. Tyto konstanty obsahují hodnoty všech parametrů, které funkce vrací, které jdou do funkcí zadávat a chybových hlášení.

Pokud ale známe hodnoty, které můžeme zadávat, nebo které funkce vrací, nemusíme konstanty využívat. Využití konstant je ale pohodlnější.

5 Canon SDK

Canon má problémy s kompatibilitou zrcadlovek a kompaktů, proto vyvíjí pro každý typ fotoaparátu jiné SDK. Bohužel některé nové verze SDK nepodporují starší fotoaparáty. Tato práce se věnuje SDK nejvyšší verzi (8.4) pro digitální zrcadlovky. Tabulky, zobrazující která verze SDK se používá pro konkrétní fotoaparát, jsou na příloženém CD. Tabulky jsou převzaté z oficiálního webového rozhraní Canon a jsou dostupné po přihlášení. Pro kompakty „SDK Canon/SDK_kompakt.html“, pro digitální zrcadlovky „SDK Canon/SDK_zrcadlovky.html“.

Podrobnosti všech instrukcí z SDK Canon jsou k nalezení v manuálech na příloženém CD (SDK Canon\RC-SDK+v8.4\ rcsdk8a.pdf, SDK Canon\RC-SDK+v8.4\ Doc\ rcsdk8b.pdf a SDK Canon\RC-SDK+v8.4\ Doc\ rcsdk8c.pdf).

5.1 Distribuce

SDK pro fotoaparáty Canon je distribuováno formou poskytnutí přístupu do vývojové sekce oficiální webové prezentace (URL: <http://www.didp.canon-europa.com/>). Veškeré funkce jsou dostupné po přihlášení.

Canon poskytuje přístup do vývojové sekce na základě formuláře dostupného na URL adrese výše v odkazu „Join Programme“, po zaškrtnutí jestli se jedná pro komerční nebo osobní použití. Přístup není povolován jednotlivcům, jen firmám nebo univerzitám (nebo lidem co patří pod firmu nebo univerzitu). Na vyřízení žádosti je 5pracovních dnů.

5.2 Použití

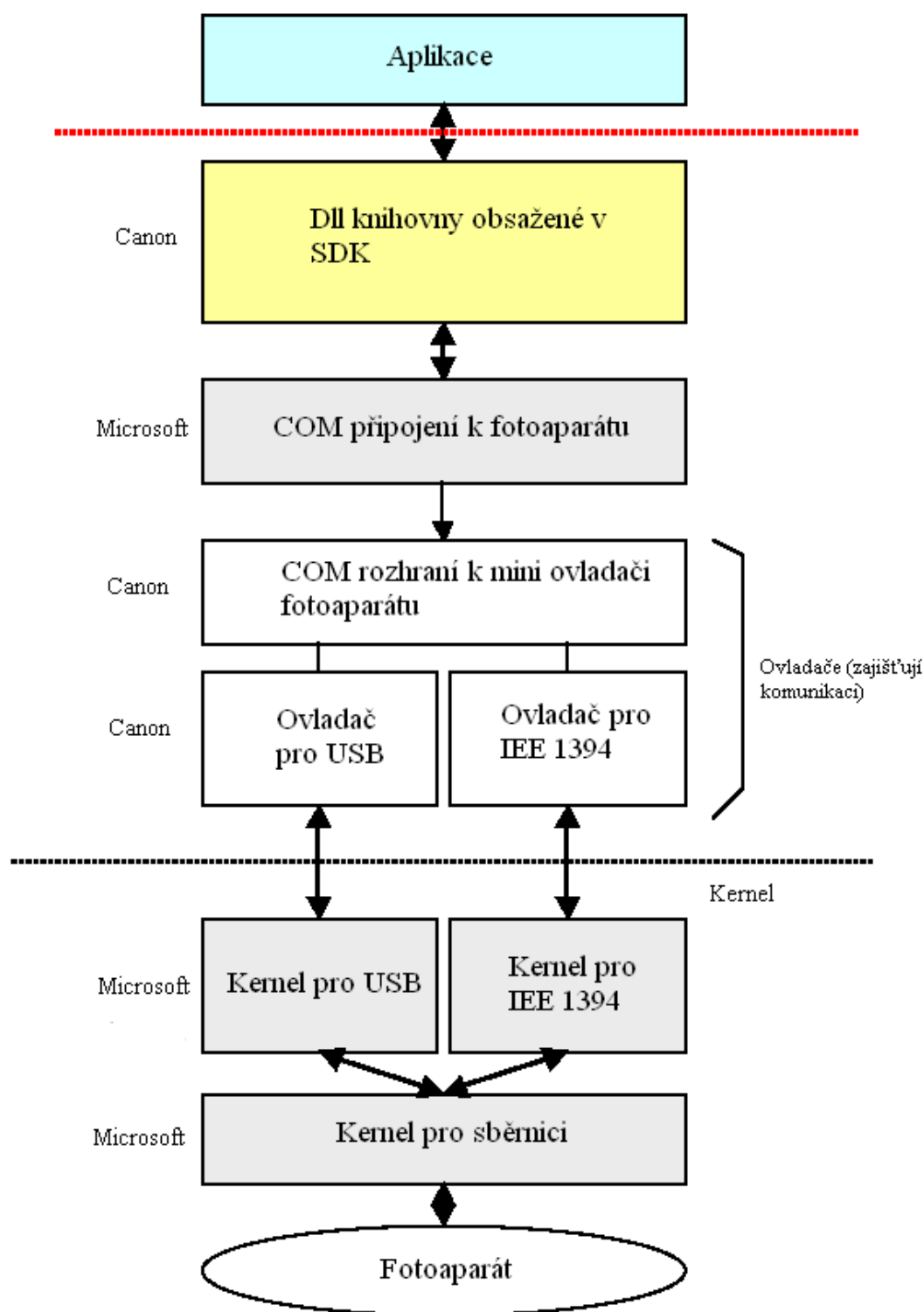
SDK Canon se nemusí instalovat (na rozdíl od Olympusu). Stačí stáhnout příslušný soubor podle typu fotoaparátu a rozbalit do vybraného adresáře. Distribuce má následující strukturu:


```
+-- Doc
| +-- Manuály ve formátu PDF
+-- Sample
| +-- Ukázkové programy v jazyku C++
+-- Win
| +-- Bin
| | +-- Dll knihovny
| +-- inc
| | +-- Soubory pro include
| +-- Lib
| | +-- Odkazy do knihoven
+-- README_WIN.TXT – popis distribuce
```

Tento vývojový nástroj je primárně navržen pro programovací jazyk C++. Proto jsou příklady, include soubory i popis v manuálu jen pro C++. Protože include soubory nejsou přepsány (jako u Olympusu), je nutno komunikovat přímo s knihovnami.

5.3 Struktura

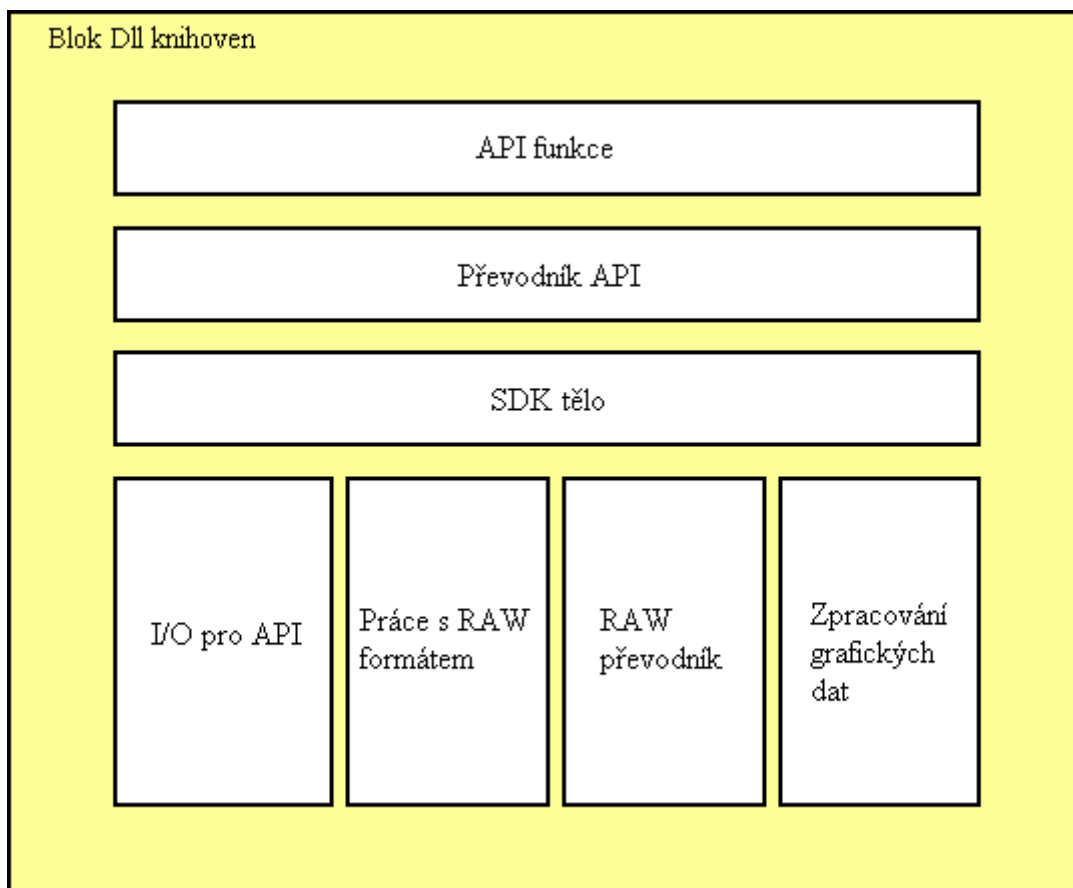
Struktura je popsána jen pro operační systém Windows XP Service Pack 2. Obrázek Obr. 4 zobrazuje blokové schéma spolupráce ovladačů Canon, ovladačů Microsoft a fotoaparátu.



Obr. 4 - Struktura SDK Canon

Pro operační systém Apple Macintosh je architektura podobná.

Blok knihoven je rozdělen do dalších několika podsystému oddělených podle funkce, jak je ukázáno na obrázku Obr. 5 .



Obr. 5 - Struktura bloku knihoven

- API funkce – jsou použitelné pro finální aplikaci
- Převodník API – převádí API funkce z formy pro aplikaci do formy pro zpracování pomocí SDK
- SDK tělo – vykonává jednotlivé instrukce
- I/O pro API – vstupní/výstupní rozhraní pro příjem/odeslání API
- Práce s RAW formátem – stará se o správné uložení
- RAW převodník – převádí RAW formát do JPEG nebo TIFF formátu
- Zpracování grafických dat – přidává do obrázku skryté informace (datum pořízení fotografie, nastavení fotoaparátu ...)

5.4 Canon SDK v Delphi

Protože SDK pro Canon je primárně určeno pro programovací jazyk C++, musíme přepsat chybové konstanty, datové typy a některé konstanty pro nastavení parametrů. Na příloženém CD jsou ve složce „SDK Canon/RC-SDK+v8.4/Win/inc“ všechny potřebné soubory pro jazyk C++. Chybové konstanty obsahuje soubor

„cdError.h“, datové typy a konstanty pro nastavení parametrů soubor „cdType.h“. Přepsané soubory jsou na přiloženém CD ve složce „SDK Canon/příklad/CamCon“ pod názvy „error.pas“, „date_type.pas“ a „constants.pas“. Jelikož jsou soubory velice obsáhlé a přepisování velice náročné na čas a znalosti obou programovacích jazyků, jsou přepsány jen konstanty a datové typy pro nejpoužívanější funkce. Některé další datové typy a konstanty jsou uvedeny v manuálu. Soubory jsou přepsány jen pro operační systém Windows.

5.5 Začátek komunikace

Před začátkem komunikace se musí aktivovat SDK. Aktivací se alokují potřebné paměťové zdroje a počítač je připraven na komunikaci s fotoaparátem. Pro aktivaci slouží funkce „*RCStartSDK(cdVersionInfo* pVersion, cdUint32 Option)*“, popsána v manuálu. Parametr „*cdVersionInfo*“ je datový typ, přes který se nastavuje verze SDK – kvůli kompatibilitě. Pro verzi, která je popisována v této práci se nastaví „*cdVersionInfo*“ takto:

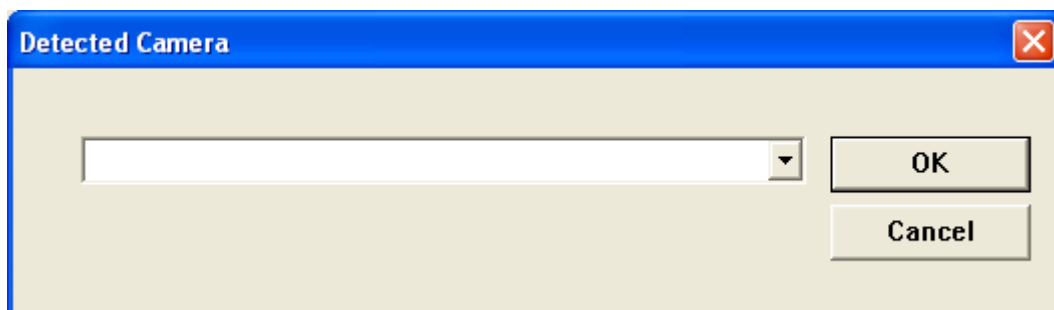
```
VersionInfo.Size:=40;  
VersionInfo.MajorVersion:=8;  
VersionInfo.MinorVersion:=0;
```

Parametr „*Option*“ musí být nastaven na 0. Je vyhrazen pro vnitřní potřeby SDK.

Před ukončením programu se musí deaktivovat SDK. Deaktivací se řádně uvolní rezervovaná paměť. Slouží k tomu funkce „*RCFinishSDK()*“ popsána v manuálu. Funkce nemá žádné parametry. V ukázkovém programu se tato funkce volá v události objektu „*Form1*“ – „*FormDestroy*“. Tím je zajištěno, že se funkce vykoná před ukončením programu.

Po aktivaci SDK můžeme začít pracovat s SDK, například funkcí pro přesné zjištění verze, pomocí funkce „*RCGetSDKVersion(cdVersionInfo*, pVersion)*“ popsána v manuálu. Zavoláním této funkce se nastaví datový typ „*cdVersionInfo*“ na přesné parametry. V přiloženém programu se aktuální verze zobrazuje v titulku okna aplikace.

Pro začátek práce s fotoaparátem se musí vybrat použitelný fotoaparát a připravit počítač na přenos dat. K tomuto je určena funkce „*RCShowDeviceSelectingUI(cdHWNDR hWndParent, cdSourceInfo* pSourceInfo)*“ popsaná v manuálu. Zavoláním této funkce se zobrazí formulář na obrázku Obr. 6. Do prvního parametru



Obr. 6- Dialog pro výběr fotoaparátu

funkce („*hWndParent*“) zadáváme handle okna, ve kterém se má dialog zobrazit (většinou aktivní okno). Do druhého parametru (datový typ „*cdSourceInfo*“) uloží SDK nastavení pro přenos dat. V dialogovém okně jsou zobrazeny jen připojené fotoaparáty a fotoaparáty, které jsou kompatibilní s verzí SDK a samozřejmě ty fotoaparáty, které podporují remote control.

Nyní se můžeme připojit k fotoaparátu funkcí „*RCOpenSource (cdSourceInfo* pSourceInfo, cdHSource* pSource)*“, kde datový typ „*cdSourceInfo*“ je získaný z výše popsané funkce pro zobrazení dialogového okna a datový typ „*cdHSource*“ obsahuje informace k identifikaci fotoaparátu.

Opět po ukončení práce se musí zavolat funkce „*RCCloseSource(cdHSource hSource)*“, která ukončí spojení s fotoaparátem. Parametrem je datový typ nastavený funkcí „*RCOpenSource*“.

5.6 Povolení/zakázání zásahu uživatele

SDK Canon podporuje velice užitečnou funkci a sice zakázání práce s fotoaparátem jinak než přes počítač, pokud je k počítači připojen. Pro některé funkce je to dokonce nezbytné. Provádí se to pomocí funkce „*RCLockUI(cdHSource hSource)*“, kde datový typ „*cdHSource*“ (získaný v kapitole 5.5) obsahuje identifikaci fotoaparátu. Funkce „*RCUnlockUI(cdHSource hSource)*“ prací s fotoaparátem připojeným k počítači opět povolí. Parametry má stejné jako funkce „*RCLockUI*“.

Pokud se funkci „*RCLockUI*“ z nějakého důvodu nepodaří vykonat (je již prováděna, nelze provést...) je opakována v intervalu nastaveném ve funkci „*RCSetUILockTimeOutTime(cdHSource hSource, cdUInt32 Time)*“. Parametr „*hSource*“ je získaný v kapitole 5.5 (identifikuje fotoaparát), parametr „*Time*“ určuje nastavovaný čas. Výchozí hodnota (pokud nebude tato funkce zavolána) je 10 vteřin. Pokud se nepodaří funkce vykonat do limitního času (určeného konstantou „*cdDEVICE_BUSY*“) je navrácena chyba.

5.7 Pořízení fotografie

Abychom mohli fotografovat pomocí SDK musíme fotoaparát přepnout do módu dálkové spouště. Tímto přepnutím se fotoaparát připraví na fotografování a bez přepnutí nebudou následující funkce fungovat. Přepnutí se dělá pomocí funkce

```
RCStartCaptureSession(  
    cdHSource hSource,  
    cdReleaseEventCallbackFunction* pFunc,  
    cdContext kontext)
```

Kde „*cdHSource*“ je datový typ identifikující fotoaparát (definovaný v kapitole 5.5), „*pFunc*“ je callback funkce (popis níže) a „*kontext*“ je parametr, který se předá callback funkci.

Callback funkce jsou funkce, které reagují na události přicházející z fotoaparátu. Abychom mohli tuto funkci používat, musíme ji registrovat událostí funkcí

```
RCRegisterEventCallbackFunction (  
    cdHSource hSource, cdEventCallbackFunction* pFunc, cdUInt32 Context,  
    cdHandle* phFunc )
```

Ve které „*hSource*“ je identifikátor fotoaparátu (získaný v kapitole 5.5), „*pFunc*“ je callback funkce (je vykonána po zavolání registrované funkce), „*Context*“ je předávaný parametr a „*phFunc*“ je ukazatel na registrovanou funkci (předává se při jejím rušení). V poslední řadě se musí definovat událost callback funkce pomocí

```
cdEventCallbackFunction(  
    cdEventID EventID,  
    const cdVoid * pData,  
    cdUInt32 DataSize,
```

cdContext Context);

Detailní popis je v manuálu.

Po nadefinování všech potřebných funkcí, které čekají na dokončení fotografování, můžeme pořídit fotografii funkcí „*cdCAPI RCCapture (cdHSource hSource)*“. Parametr „*hSource*“ je již známý identifikátor fotoaparátu. Jakmile je fotografování dokončeno je provedena funkce udaná ukazatelem „*pFunc*“ v „*RCRegisterEventCallbackFunction()*“.

Nesmíme zapomenout na návrat do normálního módu fotoaparátu pomocí funkce „*cdCAPI RCEndCaptureSession (cdHSource hSource)*“.

5.8 Ostatní funkce

V této kapitole práce popisuje funkce, které nejsou nutné k fotografování, ale umožňují snadnější práci s SDK, nebo pomáhají při programování. Většina z nich se dá velice snadno a efektivně použít v aplikaci pro zvýšení komfortu uživatele, nebo kontroly kompatibility.

Zjištění dostupných funkcí je velice užitečná funkce, která vrací datový typ se všema dostupnými funkcemi, v němž každá položka typu je datovým typem. Tento datový typ je velice obsáhlý, popisuje ho soubor pro programovací jazyk C++ umístěný na přiloženém CD ve složce „SDK Canon\RC-SDK+v8.4\Win\inc“. Vzhledem k rozsáhlosti tohoto datového typu poskytuje opravdu veškeré informace o aktuálně použitém SDK. Datový typ se naplní funkcí „*RCGetFunctions(rcFunctions* pFunctions)*“.

Užitečnou a pro získávání dalších informací jsou funkce pro získávání připojených paměťových zařízení. Jako první musíme pomocí funkce „*RCEnumVolumeReset (cdHSource hSource, cdHEnum* phEnum)*“ získat dále potřebný ukazatel „*cdHEnum*“. Nyní můžeme pomocí funkce „*RCEnumVolumeNext (cdHEnum hEnum, cdHVolume* phVolume)*“ získat ukazatel na konkrétní paměťové místo. Tato funkce se volá v cyklu, abychom získali všechny paměťová zařízení. Jakmile funkce vrátí hodnotu konstanty „*cdENUM_NA*“ je prohledávání ukončeno. Kvůli vnitřním procesům SDK je nutno po ukončení prohledávání zavolat funkci „*RCEnumVolumeRelease(cdHEnum hEnum)*“.

Nyní můžeme získat podrobné informace o paměťových zařízeních. Pomocí funkce „*RCGetVolumeCount(cdHEnum hEnum, cdUInt32* pCount)*“ můžeme získat počet nalezených paměťových modulů. Parametr „*pCount*“ je ukazatel na paměťové místo kam se počet uloží. Zajímavější funkce je „*RCGetVolumeInfo(cdHVolume hVolume, cdVolumeInfo* pVolInfo)*“, která do datového typu „*cdVolumeInfo*“ uloží informace o paměťovém modulu, daném datovým typem „*cdHVolume*“ získaného z předchozího odstavce. Datový typ „*cdHVolume*“ je popsán v manuálu a uchovává typ paměti, jestli je přepisovatelná, celkovou kapacitu, volnou kapacitu a jméno jednotky.

Po předchozím postupu máme ukazatele na všechna paměťová místa fotoaparátu. Díky tomu je můžeme prohledávat. Pro prohledávání fotografií se používá sada funkcí. Nejprve musíme zavolat funkci, která započne prohledávání : „*RCEnumImageItemReset (cdHItem hItem, cdUInt32 Depth, cdEnumItemOption Option, cdHEnum* phEnum)*“. Jako parametr „*hItem*“ použijeme „*cdHVolume*“ získaný dříve, „*Depth*“ určuje do jaké hloubky složek se bude prohledávat, „*Option*“ určuje jaké typy souborů se budou vyhledávat a „*phEnum*“ je paměťové místo, kam se uloží informace použitelné dále. Teď máme všechno připravené na prohledávání pomocí funkce „*RCEnumImageItemNext (cdHEnum hEnum, cdHItem* phItem)*“. Parametr „*hEnum*“ jsme získali výše zmíněnou funkcí a „*phItem*“ je ukazatel na paměťové místo, kam se opět uloží informace dále použitelné. Tato funkce se také většinou provádí v cyklu. Funkce opět vrací hodnotu konstanty „*cdENUM_NA*“ pokud vyhledávání skončilo. Opět po ukončení vyhledávání musíme zavolat funkci „*RCEnumImageItemRelease(cdHEnum hEnum)*“. V tomto bodě máme uloženy ukazatele na všechny soubory daného typu.

Můžeme opět získat počet nalezených záznamů pomocí funkce „*RCGetImageItemCount(cdHEnum hEnum, cdUInt32* pCount)*“, nebo informace o souboru, funkcí „*RCGetItemInfo (cdHItem hItem, cdItemInfo* pItemInfo)*“. Parametry jsou podobné jako u funkcí pro získávání informací o paměťových zařízeních. Datový typ „*cdItemInfo*“, na který odkazuje funkce „*RCGetItemInfo*“ je popsán v manuálu.

5.9 Práce s fotografií

Abychom mohli s fotografií uloženou ve fotoaparátu pracovat, musíme nejprve SDK říct, se kterou fotografií chceme pracovat. Uděláme to pomocí funkce „*RCOpenImage (cdHImageItem hImageItem)*“. Funkcí „*RCCloseImage (*

cdHImageItem hImageItem)“ práci s fotografií ukončíme. Parametr obou funkcí (*cdHImageItem*) jsme získali na konci kapitoly 5.8 (*cdHItem*) tento typ můžeme použít, protože „*cdHImageItem*“ je odvozen od „*cdHItem*“.

Pomocí funkce „*RCGetPicture (cdHImageItem hImageItem, cdHImageData* phImageData)*“ uložíme informace o uložení fotografie v plné velikosti do datového typu „*cdHImageData*“ (náhled fotografie použijeme funkci „*RCGetThumbnail (cdHImageItem hImageItem, cdHImageData* phImageData)*“ se stejnými parametry). Datový typ „*cdHImageItem*“ je stejný jako v předchozím odstavci.

Protože fotografie jsou ve fotoaparátu uloženy v RAW formátu, SDK obsahuje funkce pro převod do formátu čitelného pro počítač. Pomocí funkce „*RCCreateRAWObject(cdStgMedium* pStgMedium, cdHRAW * phRAW)*“ nejprve vytvoříme objekt pro práci s RAW formátem. První parametr (*cdStgMedium*) je datový typ, ve kterém je uloženo nastavení pro uložení fotografie. Pro naše účely stačí nastavit typ na konstantu „*cdMEMTYPE_FILE*“ (datový soubor) a název výsledného souboru (například „Foto.tif“). Z RAW formátu převádí funkce

```
RCDevelop( cdHRAW hRAW  
cdDevT_DevAction Action,  
cdDevTgtFormat tgtFormat,  
cdDevT_Dimension AreaAndRatio,  
cdStgMedium* pStgMedium,  
cdProgressCallbackFunction* pCallbackFunc,  
cdContext Context,  
cdProgressOption ProgressOption ).
```

Nyní si popíšeme jednotlivé parametry. „*cdHRAW*“ jsme definovali funkcí „*RCCreateRAWObject*“. „*cdDevT_DevAction*“ určuje (pomocí konstant) jakou chce mít výslednou fotografii, pro plné rozlišení použijeme konstantu „*kDevAct_FullDevelop*“. „*cdDevTgtFormat*“ určuje (také pomocí konstanty) výsledný typ fotografie, pro příponu .tif to je konstanta „*cdTGT_FORMAT_BMP_TOPDOWN*“. „*cdDevT_Dimension*“ je detailně popsán v manuálu, obsahuje informace o převodu. „*cdStgMedium*“ je popsán výše. „*cdProgressCallbackFunction*“ je ukazatel na callback funkci o jejíž definici pojednává kapitola 5.7. „*cdContext*“ není nikde pořádně popsán nastavuje se na 0. „*cdProgressOption*“ obsahuje nastavení pro callback funkci.

5.10 Detekce chyb

V SDK Canon se chyby detekují pomocí konstant, jejichž hodnoty vrací každá funkce. Stačí si hodnotu vrácenou funkcí uložit do proměnné a tu následně porovnávat s konstantama. V manuálu je u každé funkce uvedeno jakou chybu může vracet. Nevýhodou tohoto řešení je, že všechny datové typy se musejí předávat odkazem, nebo přímým přístupem do paměti. Výhodnější je řešení od Olympusu, kdy se chyby vrací výjimkami i proto, že když nastane chyba program nemůže pokračovat dále.

6 Závěr

Všechny SDK jsou psány v C++ proto je použití programovacího jazyku Delphi nevhodné. Jen SDK Olympus obsahuje přepsané třídy do Delphi, ale nikoliv od programátorů Olympus, ale od vývojářů v Delphi, kteří se SDK Olympus zakoupili a celou strukturu přepsali do Delphi. Při použití programovacího jazyku C++ zvládne naprogramovat aplikaci se základními funkcemi průměrný programátor.

6.1 SDK Olympus

SDK je zaměřeno na komfort programátora, díky tomu že je celkem dlouho na trhu dají se na internetových forech, nebo na fóru přímo ve vývojářském centru Olympusu, najít veškeré potřebné informace. Manuál k SDK je také velice přehledně a strukturovaně napsán. Pro použití v Delphi je SDK kompletně přepsáno z C++, takže naprogramování aplikace zvládne středně pokročilý programátor v Delphi. Důsledkem toho, že Olympus začal vzdálené řízení podporovat celkem brzy, můžeme vzdáleně ovládat téměř všechny fotoaparáty na trhu. SDK má jednotnou verzi, přes kterou můžeme ovládat i starší fotoaparáty, což je bezkonkurenční výhoda.

6.2 SDK Canon

SDK Canon již nepodporuje tolik funkcí jako SDK Olympus. Hodně funkcí se zabývá převodem RAW formátu. Získat SDK od Canonu je o poznání snažší než u Olympusu. Díky tomu, že je toto řešení na trhu relativně krátce, není přepsáno do Delphi. Veškeré funkce, konstanty a datové typy je nutné přepsat z programovacího jazyka C++ do Delphi což je velice obtížné. Proto programování pomocí SDK Canon v Delphi zvládnou zkušení programátoři až profesionálové. Manuál je rozdělen do tří částí a je pěkně strukturovaně a přehledně napsaný.